Various modifications to languages that make another reg lang - this gives us a way to know some language is regular that doesn't rely on directly constructing an NFA/reg exp/right-lln grammar.

1) Given a language $L$ (on an alphabet $\Sigma$), the language $\bar{L} = \{ \text{all strings on } \Sigma \text{ not in } L \}$ is also regular.

Pf: Since $L$ is regular, we have a DFA $M$ that accepts $L$. If $M = (Q, \Sigma, \delta, q_0, F)$, then $\bar{M} = (Q, \Sigma, \delta, q_0, Q \setminus F)$ accepts $\bar{L}$. Everything in $Q \setminus F$ not in $F$.
2) Given two languages \( L_1 \) and \( L_2 \), 
\( L_1 \cup L_2 \) is regular.

**Pf:** If we have regular expressions \( e_1 \) and \( e_2 \) 
for \( L_1 \) and \( L_2 \), then \( e_1 + e_2 \) is a regular 
expression for \( L_1 \cup L_2 \).

3) Given languages \( L_1 \) and \( L_2 \), \( L_1 \cap L_2 \) 
is regular.

**Pf:** Since \( L_1 \) and \( L_2 \) are regular, we have 
DFA's \( M_1 = (Q_1, \Sigma, S_1, q_0^1, F_1) \) 
and \( M_2 = (Q_2, \Sigma, S_2, q_0^2, F_2) \) 
for them.
New machine for $L_1 \cap L_2$:

$M_1$

$M_2$

$ababaq$
Then the DFA

\[ M_1 \times M_2 = (Q_1 \times Q_2, \Sigma, S, (q_0^1, q_0^2), F_1 \times F_2) \]

where

\[ S((q_1, q_2), \sigma) = (S_1(q_1, \sigma), S_2(q_2, \sigma)) \]

states in "\( M_1 \times M_2 \)" look like \((q_1, q_2)\), where \( q_1 \in Q_1, q_2 \in Q_2 \), will accept precisely \( L_1 \cap L_2 \).

**Def:** A "homomorphism" from an alphabet \( \Sigma \) to an alphabet \( \Gamma \) is a function from \( \Sigma^* \rightarrow \Gamma^* \) given by replacing every letter in a string on \( \Sigma^* \) by a fixed string in \( \Gamma^* \).
\[ \Sigma = \{ a, b, c \} \]
\[ \Gamma = \{ 0, 1, 2 \} \]

\[ h : \Sigma^* \rightarrow \Gamma^* \] given by
- replacing all \( a \)'s with \( 01 \)'s,
- all \( b \)'s with \( 12 \)'s,
- and all \( c \)'s with \( 2 \)'s.

So

\[ h(abcabc) = 011212 \]
\[ h(baabcab) = 1201011201 \]

This \( h \) is a homomorphism.

Given a reg. lang. \( L \) and a homomorphism \( h \), then

\[ h(L) = \{ h(w) \mid w \in L \} \]

is regular.

Pf: Given a reg. expr. \( e \) for \( L \),
\( h(e) \) is a reg. expr. for \( h(L) \).
Let $L$ be a language on $\{a, b, c\}$. Consider the language on $\{b, c\}$ given by all the strings that can be obtained by deleting all the $a$'s from a string in $L$. This is regular.

**Pf:** Take an NFA for $L$ and replace all the $a$'s with $\lambda$'s.

Alternate **Pf:** consider a homomorphism $h$ where we replace $a$'s with $\lambda$'s, $b$'s with $b$'s, $c$'s with $c$'s. Then this lang. is $h(L)$. 
Let $L$ be a language on $\{a, b, \lambda\}$. Consider $d(L) = \{\text{all strings obtained by deleting a single } a \text{ from a string in } L\}$. Then $d(L)$ is regular.

\textbf{Pf}: Given an NFA $M$ for $L$.

\textbf{Example for thought:}
modify M by

1) Make a second copy of M, called M'.

2) For every $\delta \alpha$-transition from $q_1$ to $q_2$ in M, make a $\gamma$ transition from $q_1$ to copy-of-$q_2$ in M in M'.

3) The initial state is the initial state in M; the final states are the final states in M' (but not those in M).